Energy Efficiency of Buildings: Final Report

Hamza Javed (33%), Howard Mach(33%), Thompson Pham(33%)

Table of Contents

Background & Objective

We plan to use Principal Component Analysis. Principal Component Analysis is a statistical approach that is able to reduce the dimensionality of the data and capture the most important information. By doing this, we can visualize our data easier by highlighting the most important features.

After PCA we will attempt to use Multiple Linear Regression. We will discuss more in depth about it later, but we had a lot of issues implementing this model. Multiple Linear Regression models the relationship between the magnitude of our predictors and our response. With this information we can make future predictions, or in our case find the best set of feature values that minimizes our response.

Since Multiple Linear Regression did not work, we turned our efforts towards Principal Component Regression. PCR can help mediate issues with multicollinearity. It is similar to Multiple Linear Regression because they both use least squares to fit a linear regression model. Except in this case, our predictors are linear combinations of the data.

In modern history, global warming has become a larger than life issue that impacts millions of people across the globe. An increase in fossil fuel use has clouded our atmosphere resulting in hotter temperatures around the world. In 2022, Japan suffered a massive heat wave which caused the country's overall electricity use to skyrocket. In order to accommodate for the hot weather, households and businesses have increased their AC usage. Due to many buildings being decades old, Britain's overall use of heating and cooling increases year by year. The poorly insulated structures can not cope with the rising temperatures. We see a desperate need to find a resolution to high electricity use. Our project utilizes a dataset which records many metrics about the structure of a building and its AC and heating use. If we can find a model that minimizes the AC or heating use, we can recommend construction companies to build with this information in mind and reduce electricity use.

Dataset Information

We retrieved the dataset from UC Irvine's Machine Learning Repository, donated by Athanasios Tsanas and Angeliki Xifara. It is composed of a collection of 768 observations made up of 8 features. Of those 768 observations there are 192 unique observations that are copied across 4 different orientations.

Predictor Variables

- Relative Compactness: Real Number - Ratio of compactness of building to that of an equivalent rectangle
- Surface Area (m2): Real Number - Total surface area of the walls of the building in square meters
- Wall Area (m2): Real Number - Total surface area of the roof of the building in square meters
- Roof Area (m2): Real Number
- Overall Height (m): Real Number - The tallest point of the building to the ground floor
- Orientation (2:North, 3:East, 4:South, 5:West): Integer - four different directions of which the given object is facing
- Glazing Area (% of wall is glass): Real Number - the general percentage of a surface is composed of glass
- Glazing Area Distribution (glass windows/doors): Integer - The amount of glass related objects in the form of windows/door seen on a surface

Target Variables

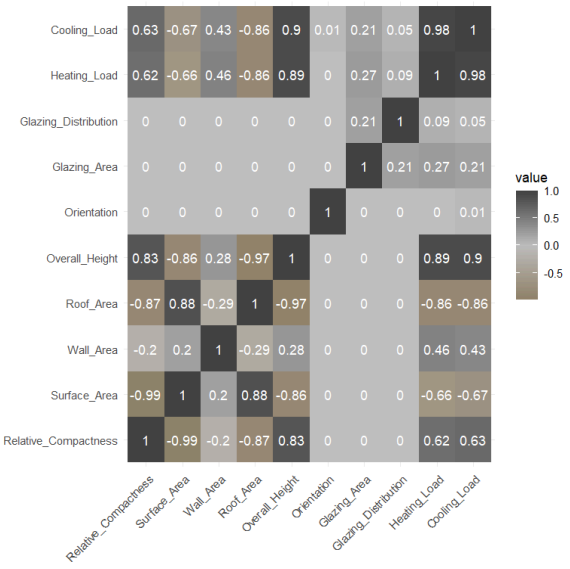
- Heating Load (kWh/m²): Real Number - The amount of energy required to heat the building per square meter
- Cooling Load (kWh/m²): Real Number - The amount of energy required to cool the building per square meter

Data Cleaning

Our dataset consisted entirely of numerical variables, eliminating the need for any one-hot encoding. Fortunately after checking for null values in our data we found none. But after seeing how the features are measured across different metrics, we thought it would be reasonable to scale our features, allowing for each one to have a similar weight. This will help with managing the data and allowing for each feature to contribute equally to the model.

Exploratory Analysis

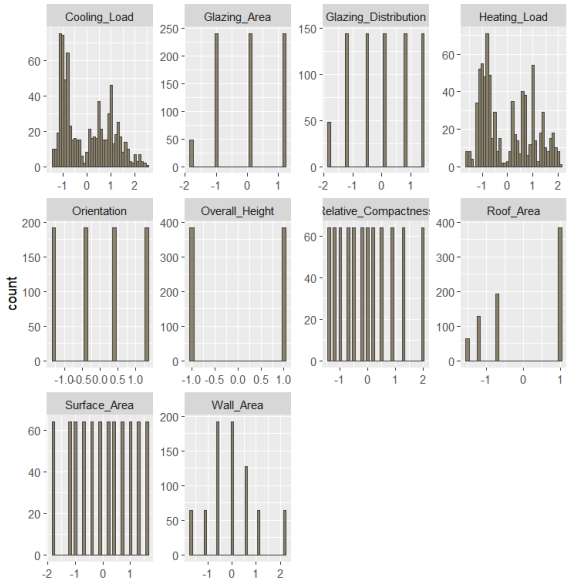
Correlation



From the correlation matrix plot, we can see that many predictors are linearly independent from each other. However, there are some predictors that are linearly dependent on each other. We will investigate this later.

Distribution

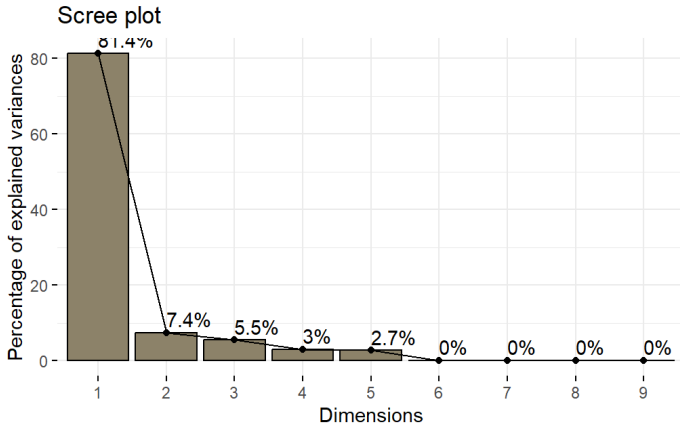
Since our data is simulated, many of the variable distributions are uniform. Our scale function does not alter the content of our data, so we should expect the spread of the data to be the same before and after.



Principal Component Analysis

After ensuring our data is clean and not missing any values, we moved on to principle analysis. Our goal is to find the most influential predictors with respect to the response. Before we can do that we should perform analysis. In our project, we used the princomp() function to compute the PCs. Next, we used the fviz_eig() function to visualize the principle components importance.

```
Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4     Comp.5      Comp.6       Comp.7       Comp.8       Comp.9
Standard deviation     1.4472190 0.43542377 0.37548012 0.2780051 0.26561131 0.034897099 1.008037e-02 2.452010e-04 6.198501e-09
Proportion of Variance 0.8136357 0.07365221 0.05476907 0.0300239 0.02740656 0.000473086 3.947431e-05 2.335641e-08 1.492570e-17
Cumulative Proportion  0.8136357 0.88728788 0.94205695 0.9720809 0.99948742 0.999960502 1.000000e+00 1.000000e+00 1.000000e+00
```

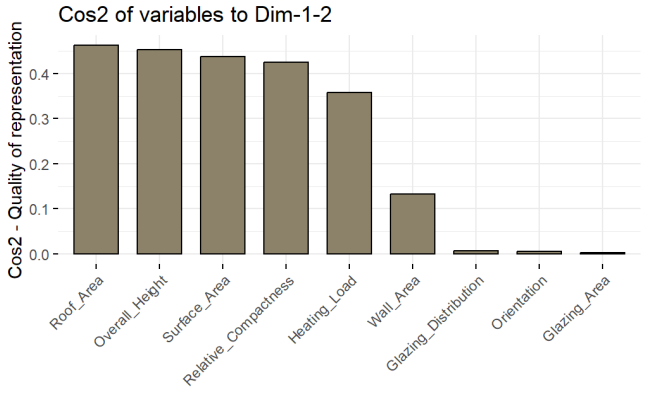


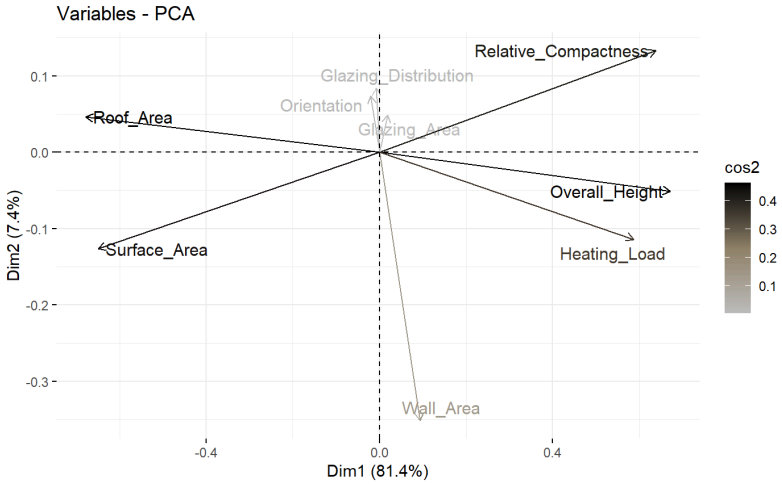
From the figure above the cumulative proportion of each component is a good indicator as to how many principal components we need. We want this value to be in the range 80-90. Component 1 has 81 percent and component 2 has 88 percent, so it is sufficient to say that the first two principal components accurately represent the data. We found the following loadings associating the variables and the components

	Comp.1	Comp.2
Relative_Compactness	0.44106085	0.3051457
Surface_Area	-0.44868782	-0.2901866
Wall_Area	0.06469574	-0.8067578
Roof_Area	-0.46877808	0.1066577
Overall_Height	0.46400686	-0.1171952
Orientation	-0.01429221	0.1684887
Glazing_Area	0.01326820	0.1115106
Glazing_Distribution	-0.00568764	0.1932131
Heating_Load	0.40557505	-0.2618298

From the figure above, the magnitude of the loading indicates the variable's contribution to the component. The sign indicates whether the variable is positively or negatively correlated to the principal component.

The loadings by themself are a bit confusing and hard to imagine, so we created a biplot to visualize our findings. Before we did that we used a Cos^2 metric on our data by using fviz_cos2(). Through research, I found it was a great method to use this metric in variable importance. The following visual explains the influence each variable has on the principal components.

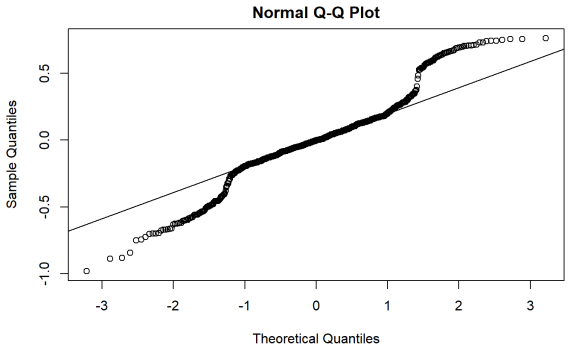


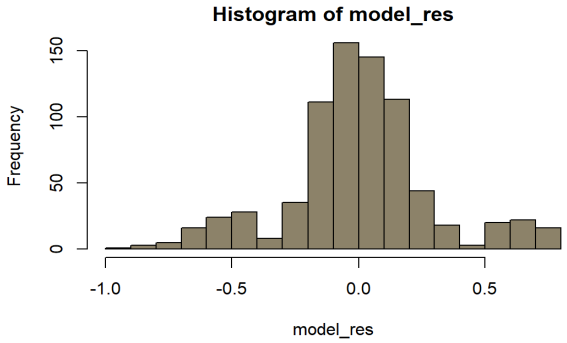


From the figures above we can see that our most influential variables are Roof_Area, Overall_Height, Surface_Area, and Relative_Compactness. From the biplot, all of the variables that are near each other are highly correlated to each other while variables opposite to them are lowly correlated to them. The magnitude of the vectors indicates its importance to the model. With this information in mind, we created a Multiple Linear Regression model.

Multiple Linear Regression

Assumptions





When delving into multiple linear regression, we turned towards histograms and Normal Q-Q plot to build our general assumptions on the potential model. We had to make sure that the residuals are somewhat linear and are normally distributed. After we applied those two plots we found that overall, the dataset is quite linear but there is a slight curve in the Q-Q plot at the tail. Alongside this we saw with our histogram, we saw a slight dip at 25% and 75%, but for the most part it was a normal distribution. These two given instances show to us some signs that the dataset has outliers. But all in all, based on the given assumptions the dataset is sufficient to be applied with multiple linear regression.

ANOVA

```
Call:
lm(formula = Heating_Load ~ Relative_Compactness + Surface_Area +
    Wall_Area + Roof_Area + Overall_Height + Orientation + Glazing_Area +
    Glazing_Distribution, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9808 -0.1308 -0.0025  0.1341  0.7636

Coefficients: (1 not defined because of singularities)
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -1.655e-15  1.049e-02   0.000  1.00000
Relative_Compactness -6.790e-01  1.079e-01  -6.295 5.19e-10 ***
Surface_Area         -7.620e-01  1.491e-01  -5.112 4.04e-07 ***
Wall_Area             2.629e-01  2.874e-02   9.148  < 2e-16 ***
Roof_Area                    NA         NA      NA       NA
Overall_Height        7.237e-01  5.866e-02  12.338  < 2e-16 ***
Orientation          -2.587e-03  1.050e-02  -0.246  0.80548
Glazing_Area          2.632e-01  1.075e-02  24.488  < 2e-16 ***
Glazing_Distribution  3.132e-02  1.075e-02   2.915  0.00367 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2908 on 760 degrees of freedom
Multiple R-squared:  0.9162,    Adjusted R-squared:  0.9154
F-statistic:  1187 on 7 and 760 DF,  p-value: < 2.2e-16
```

Heating_Load(Full Model)

```
Call:
lm(formula = Cooling_Load ~ Relative_Compactness + Surface_Area +
    Wall_Area + Roof_Area + Overall_Height + Orientation + Glazing_Area +
    Glazing_Distribution, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.91388 -0.16404 -0.02804  0.14682  1.17493

Coefficients: (1 not defined because of singularities)
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -1.229e-15  1.214e-02   0.000    1.000
Relative_Compactness -7.871e-01  1.248e-01  -6.306 4.85e-10 ***
Surface_Area         -8.171e-01  1.725e-01  -4.737 2.59e-06 ***
Wall_Area             2.049e-01  3.326e-02   6.161 1.17e-09 ***
Roof_Area                    NA         NA      NA       NA
Overall_Height        7.885e-01  6.787e-02  11.618  < 2e-16 ***
Orientation           1.429e-02  1.215e-02   1.176    0.240
Glazing_Area          2.061e-01  1.244e-02  16.573  < 2e-16 ***
Glazing_Distribution  6.635e-03  1.244e-02   0.534    0.594
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3365 on 760 degrees of freedom
Multiple R-squared:  0.8878,    Adjusted R-squared:  0.8868
F-statistic: 859.1 on 7 and 760 DF,  p-value: < 2.2e-16
```

Cooling_Load(Full Model)

At the start when we were building the full model, we found that there can only be one target variable. So we built two models, one with heating_load and the other with cooling_load. Afterwards we compared the two using Rstudio’s summary function and

found that heating_load edged out cooling_load with a better R-squared value. This means that heating_load in contrast to cooling_load when building a model off of fits the dataset just marginally better. After choosing the target variable, we made use of stepwise variable selection to help remove variables that were not contributing significantly to the model's predictive power. By the end of it, we removed roof_area and orientation from our full model leaving us with our reduced model to analyze and test.

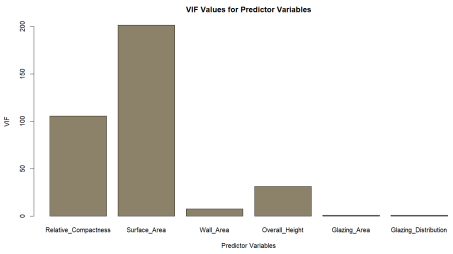
After applying the summary function from rStudio, we see that the reduced model was able to take care of some problem variables and improve our r-squared value marginally.

```
lm(formula = Heating_Load ~ Relative_Compactness + Surface_Area +
    Wall_Area + Overall_Height + Glazing_Area + Glazing_Distribution,
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9843 -0.1307 -0.0026  0.1346  0.7648

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         -1.655e-15  1.049e-02   0.000  1.00000
Relative_Compactness -6.790e-01  1.078e-01  -6.299 5.06e-10 ***
Surface_Area         -7.620e-01  1.490e-01  -5.115 3.97e-07 ***
Wall_Area             2.629e-01  2.873e-02   9.153  < 2e-16 ***
Overall_Height        7.237e-01  5.862e-02  12.345  < 2e-16 ***
Glazing_Area          2.632e-01  1.074e-02  24.503  < 2e-16 ***
Glazing_Distribution  3.132e-02  1.074e-02   2.916  0.00365 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2906 on 761 degrees of freedom
Multiple R-squared:  0.9162,    Adjusted R-squared:  0.9155
F-statistic:  1387 on 6 and 761 DF,  p-value: < 2.2e-16
```

When we arrived at our desired reduced model, we compared it to the full model using ANOVA to assess if it was sufficient enough of a change that would affect the overall output. With Pr(>F) exceeding over 0.05, we found that the reduced model did lead to statistical differences compared to the full model. Therefore, we can conclude that the reduced model adequately captures the variations in the data.

```
Analysis of Variance Table

Model 1: Heating_Load ~ Relative_Compactness + Surface_Area + Wall_Area +
    Roof_Area + Overall_Height + Orientation + Glazing_Area +
    Glazing_Distribution
Model 2: Heating_Load ~ Relative_Compactness + Surface_Area + Wall_Area +
    Overall_Height + Glazing_Area + Glazing_Distribution
  Res.Df    RSS Df  Sum of Sq      F Pr(>F)
1    760 64.273
2    761 64.278 -1 -0.0051323 0.0607 0.8055
```

```
Relative_Compactness         Surface_Area            Wall_Area
           105.524054           201.531134             7.492984
      Overall_Height         Glazing_Area Glazing_Distribution
           31.205474             1.047508             1.047508
```



Before concluding anything, we ran one last test (VIF) to see if the reduced model may experience any problems. And after running the test we saw that three of the variables (Relative_Compactness, Surface_Area, Overall_Height) had extremely high VIF values. This indicated issues with multicollinearity, making the general model very sensitive to noise. As a result, we felt we couldn't go further with multiple linear regression and turned towards an alternative that was more adept at dealing with the current problems.

Principal Component Regression

We used the pcr() function to create our model. The summary is the figure below. Similar to PCA we want to choose the number of components based on the proportion of variance explained. We can also choose the number of components based on the RMSE value. We can see that principal components 1, 2, and 3 have a low adjCV

value. These three components also explain 86 percent of the variance. It is sufficient to say that these three components accurately represent the data. The coefficients the model provides is listen below. From the coefficients, we can see that we can see that Wall_Area, Roof_Area, and Overall_Height are our most influential predictors on our target variable.

```
Data:   X dimension: 768 8
        Y dimension: 768 1
Fit method: svdpc
Number of components considered: 8

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
CV           1.001   0.6080   0.4528   0.3745   0.3752   0.3451   0.2949   0.2922   0.2940
adjCV        1.001   0.6079   0.4469   0.3734   0.3750   0.3450   0.2947   0.2920   0.2921

TRAINING: % variance explained
             1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X              46.29    61.78    76.95    89.45    99.28    99.94   100.00   100.00
Heating_Load   63.23    80.87    86.13    86.13    88.24    91.46    91.62    91.63
```

	Heating_Load
Relative_Compactness	0.11162513
Surface_Area	-0.11935406
Wall_Area	0.34640489
Roof_Area	-0.28368563
Overall_Height	0.28381365
Orientation	-0.02128755
Glazing_Area	0.13950770
Glazing_Distribution	0.16415721

Problems

Some of the problems we had with our data are the multicollinearity and the simulated nature of our data. As discussed earlier, this issue forced us to switch to another model for finding most influential predictors. Also, our data was simulated so that means our predictors have different relationships with the target variables and that could also cause some of the issues with multicollinearity.

Future Work

In the future, we could implement our model with real world data sets. If we used real-world data, the analysis wouldn't be based on assumptions we had with the simulated data. We might discover new variables that influence our models, and the interactions between variables could be more complex and unexpected. Real-world

data could also reveal more biases or limitations, leading to a more accurate understanding of how pollution happens.

Beyond linear models like multiple regression and PCR, we can explore tree-based models like random forests or gradient boosting. These models don't require assumptions about linear relationships and can handle complex interactions between variables, providing a more flexible way to identify the most influential factors.

References

Keita, Zoumana. “Principal Component Analysis (PCA) in R Tutorial.” *DataCamp*, DataCamp, 13 Feb. 2023, www.datacamp.com/tutorial/pca-analysis-r.

Zach Bobbitt. “Principal Components Regression in R (Step-by-Step).” *Statology*, 16 Nov. 2020, www.statology.org/principal-components-regression-in-r/.

“Energy Efficiency.” *UCI Machine Learning Repository*, archive.ics.uci.edu/dataset/242/energy+efficiency. Accessed 9 May 2024.

Code:

Script 1)

```{r}

library(readxl)
library(ggplot2)
library(tidyr)
library(reshape2)

data <- read_excel("ENB2012_data.xlsx")
data <- as.data.frame(scale(data))

str(data)

```

```{r}

is.null(data$X1)
is.null(data$X2)
is.null(data$X3)
is.null(data$X4)
is.null(data$X5)
is.null(data$X6)
is.null(data$X7)
is.null(data$X8)
is.null(data$Y1)
is.null(data$Y2)
```

```{r}
# using Heating Load

fit_data_x1 <- lm(data = data, Y1~X1)
fit_data_x2 <- lm(data = data, Y1~X2)
fit_data_x3 <- lm(data = data, Y1~X3)
fit_data_x4 <- lm(data = data, Y1~X4)
```

```
fit_data_x5 <- lm(data = data, Y1~X5)
fit_data_x6 <- lm(data = data, Y1~X6)
fit_data_x7 <- lm(data = data, Y1~X7)
fit_data_x8 <- lm(data = data, Y1~X8)


par(mfrow=c(2,3))
plot(data$Y1~data$X1, main = "Compactness")
abline(fit_data_x1)

plot(data$Y1~data$X2, main = "Surface Area")
abline(fit_data_x2)

plot(data$Y1~data$X3, main = "Wall Area")
abline(fit_data_x3)

plot(data$Y1~data$X4, main = "Roof Area")
abline(fit_data_x4)

plot(data$Y1~data$X5, main = "Overall Height") # not significant
abline(fit_data_x5)

plot(data$Y1~data$X6, main = "Orientation") # not significant
abline(fit_data_x6)

plot(data$Y1~data$X6, main = "Glazing Area") # not significant
abline(fit_data_x7)

plot(data$Y1~data$X6, main = "Glazing Distribution") # not significant
abline(fit_data_x8)


```

```{r}
# Using Cooling load

fit_data_x1 <- lm(data = data, Y2~X1)
fit_data_x2 <- lm(data = data, Y2~X2)
fit_data_x3 <- lm(data = data, Y2~X3)
```

```
fit_data_x4 <- lm(data = data, Y2~X4)
fit_data_x5 <- lm(data = data, Y2~X5)
fit_data_x6 <- lm(data = data, Y2~X6)
fit_data_x7 <- lm(data = data, Y2~X7)
fit_data_x8 <- lm(data = data, Y2~X8)


par(mfrow=c(2,3))
plot(data$Y2~data$X1, main = "Compactness")
abline(fit_data_x1)

plot(data$Y2~data$X2, main = "Surface Area")
abline(fit_data_x2)

plot(data$Y2~data$X3, main = "Wall Area")
abline(fit_data_x3)

plot(data$Y2~data$X4, main = "Roof Area")
abline(fit_data_x4)

plot(data$Y2~data$X5, main = "Overall Height") # not significant
abline(fit_data_x5)

plot(data$Y2~data$X6, main = "Orientation") # not significant
abline(fit_data_x6)

plot(data$Y2~data$X6, main = "Glazing Area") # not significant
abline(fit_data_x7)

plot(data$Y2~data$X6, main = "Glazing Distribution") # not significant
abline(fit_data_x8)

```


```{r}
library(readxl)
library(ggplot2)
data_temp <- read_excel("ENB2012_data.xlsx")
data_temp <- as.data.frame(scale(data_temp))
data_temp <- subset(data_temp, select = -Y2)
```

```
colnames(data_temp) <- c("Relative_Compactness", "Surface_Area", "Wall_Area",
"Roof_Area", "Overall_Height", "Orientation", "Glazing_Area", "Glazing_Distribution",
"Heating_Load")


data_long <- tidyr::gather(data, key = "variable", value = "value")

ggplot(data_long, aes(x = value)) +
  geom_histogram(fill = "#908269", color = "#434343", binwidth = 0.1) +
  facet_wrap(~ variable, scales = "free") +
  labs(title = "Distribution of Variables")

ggplot(data_long, aes(x = variable, y = value)) +
  geom_boxplot() +
  labs(x = "Attribute", y = "Value", title = "Boxplot for Each Attribute") +
  facet_wrap(~variable, scales = "free")

# Correlation Matrix
cor_matrix <- cor(data)
cor_matrix_long <- melt(cor_matrix)

ggplot(cor_matrix_long, aes(Var1, Var2)) +
  geom_tile(aes(fill = value), color = "NA") +
  geom_text(aes(label = round(value, 2)), color = "white") +  # Add correlation values as
text
  scale_fill_gradient2(low = "#908269", mid = "grey", high = "#434343", midpoint = 0,
na.value = "grey50") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Correlation Matrix Plot")


data_long <- tidyr::gather(data, key = "variable", value = "value")

ggplot(data_long, aes(x = value)) +
  geom_histogram(fill = "#908269", color = "#434343", binwidth = 0.1) +
  facet_wrap(~ variable, scales = "free") +
  labs(title = "Distribution of Variables")
```

```

```{r}
#PCA

library(FactoMineR)
library(ggcorrplot)
library(corrr)
library(factoextra)

corr_mat <- cor(data_temp)
ggcorrplot(corr_mat,type = 'lower', hc.order = T, lab = T)

cat(" ")

data_pca <- princomp(corr_mat)
summary(data_pca)

#from the summary(data_pca) the first principle component explains about 85% fo the total variance. We can say about 85% of the data can be represented in the first principal component. The second principle component explains about 6 % of the total variance. From this data, we conclude that the first two (mostly the first) accurately represent the data.

data_pca$loadings[,1:2]

# The magnitude of the loadings indicate the variables contribution to the component. The sign of the loadings indicates wether the variable is positively or negativley correlated to the principal component.

fviz_eig(data_pca, addlabels = T, barfill = "#908269", barcolor = "black")

# visualizing the principle components importance

fviz_pca_var(data_pca, col.var = "black")

# variables that are grouped together are positively correlated to each other. The magnitude of the distance from the variable to the origin indicates how well the variable is represented.
```
```

fviz_cos2(data_pca, choice = "var", axes = 1:2, fill = "#908269", color = "black")

low values indicate that the variable is not well represented by the first two principle components. A high value is the opposite. cos^2 is a good metric to use for this instance.
Roof Area, Overall Height, and Surface Area are the top three variables with the highest Cos^2 value, hence contributing the most to Principal component 1 and 2.

fviz_pca_var(data_pca, col.var = "cos2",
             gradient.cols = c("grey" , "#908269", "black"),
             repel = TRUE)

Biplot with the color representing the cos^2 value.

```

```{r}

linear_r_model <- lm(Heating_Load ~ Relative_Compactness + Surface_Area + Wall_Area + Roof_Area + Overall_Height + Orientation + Glazing_Area + Glazing_Distribution, data = data_temp)

linear_r_model <- lm(Cooling_Load ~ Relative_Compactness + Surface_Area + Wall_Area + Roof_Area + Overall_Height + Orientation + Glazing_Area + Glazing_Distribution, data = data_temp)

linear_r_model <- lm(cbind(Heating_Load, Cooling_Load) ~ Relative_Compactness + Surface_Area + Wall_Area + Roof_Area + Overall_Height + Orientation + Glazing_Area + Glazing_Distribution, data = data_temp)
summary(linear_r_model )

```

```{r}
library(car)

lin_model <- lm(Heating_Load ~ Relative_Compactness + Surface_Area + Wall_Area + Roof_Area + Overall_Height + Orientation + Glazing_Area + Glazing_Distribution, data = data_temp)

```
# check normality assumption
model_res <- lin_model$residuals

hist(model_res, breaks = 20, col = "#908269")
# histogram indicates that the residuals are mostly normal with some skew to the left

qqnorm(model_res)
qqline(model_res)
# qqplot of residuals indicates that the residuals mostly do not follow a normal distribution. we know this since many of the points do not fall on the line.

next we will find the best model using the data from PCA. we will build a model without orientation, wall area, glazing area, and glazing distribution

lin_model_2 <- lm(Heating_Load ~ Relative_Compactness + Surface_Area + Wall_Area + Overall_Height + Glazing_Area + Glazing_Distribution, data = data_temp)

anova(lin_model, lin_model_2)

vif(lin_model_2)

```

```{r}
#stepwise subset

int_only <- lm(data = data_temp, Heating_Load  ~ 1)

all <- lm(Heating_Load ~ Relative_Compactness + Surface_Area + Wall_Area + Roof_Area + Overall_Height + Orientation + Glazing_Area + Glazing_Distribution, data = data_temp)

both <- step(int_only, direction = 'both', scope = formula(all), trace = 0)

both$anova

both$coefficients
```

```

```{r}
library(pls)

set.seed(123)

pcr_model <- pcr(Heating_Load ~ Relative_Compactness + Surface_Area + Wall_Area
+ Roof_Area + Overall_Height + Orientation + Glazing_Area + Glazing_Distribution,
data = data_temp, validation = "CV")

summary(pcr_model)

validationplot(pcr_model, val.type="MSEP")


# split data for predictions
# split 70% for training and 30% for test

sample <- sample(c(TRUE, FALSE), nrow(data_temp), replace=TRUE, prob=c(0.7,0.3))

train_data <- data_temp[sample, ]
temp <- data_temp[!sample, ]
test_data <- subset(temp, select = -Heating_Load)
test_data_y <- subset(temp, select = Heating_Load)


model_train <- pcr(Heating_Load ~ Relative_Compactness + Surface_Area +
Wall_Area + Roof_Area + Overall_Height + Orientation + Glazing_Area +
Glazing_Distribution, data = train_data, validation = "CV")

predict_pcr <- predict(model_train, test_data, ncomp = 2)

sqrt(mean((as.numeric(predict_pcr) - test_data_y$Heating_Load)^2))

mod_summ <- summary(model_train)

model_train$coefficients

```
```